ABSTRACT

With the advent of networked computer systems that connect disparate computer hardware and operating systems, it is important for port simulation systems to be able to run on a wide variety of computer platforms. This paper describes the design and implementation issues in reengineering the PORTSIM model in order to field the model to Windows-based systems as well as to Unix-based systems such as the Sun, Silicon Graphics, and HP workstations. The existing PORTSIM model was written to run on a Sun workstation running Unix. The model was initially implemented in MODSIM and C and utilized embedded SQL to retrieve port, ship, and cargo data from back-end ORACLE databases. Output reports, graphs, and tables for model results were written in C, utilizing third-party graphics libraries. This design and implementation worked well for the intended hardware platform and configuration, but as the number of model users grew and as the capabilities of the model expanded, a need developed to field the model to varying hardware configurations. This new requirement demanded that the existing design be modified to more easily allow for model fielding and maintenance. A phased approach is described that (1) identifies the existing model from which cross-platform development began, (2) delineates an intermediate client-server model that has been developed utilizing Java to allow for greater flexibility and ease in distributing and fielding the model, and (3) describes the final goals to be achieved in this development process.

KEYWORDS
Maritime, seaport, logistics, transportation, PORTSIM, MODSIM, Java, client-server

1. Introduction
The ability to field simulation systems to a large number of sites is of major importance to many simulation tool users. These sites typically consist of computer platforms that are varied in both hardware and operating systems. It is highly desirable for a simulation system to be easily fielded to these many sites. The Port Simulation System (PORTSIM) [1,2,3,4], a seaport simulation model developed by Argonne National Laboratory,
DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
DISCLAIMER

 Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
provides an example of how the design of an existing model can be modified to shorten the development cycle for delivery of the system to multiple computer systems. The time saved in the development cycle benefits both developers and users of the simulation system. Developers gain by utilizing a facilitated procedure to make the delivery of the system a success. Users of the simulation system benefit because the system is delivered in a shorter time frame with fewer restrictions on computer hardware and software requirements.

2. Model Description
PORTSIM is a discrete-event, time-stepped simulation that facilitates the analysis of movements of military unit equipment through worldwide seaports and allows for detailed infrastructure analysis. PORTSIM assists planners in comparing and selecting ports and it determines port throughput capability and utilization of critical resources. PORTSIM is designed to answer the following questions:

- How long does it take to move equipment and supplies through the seaport (closure)?
- What and where are the potential bottlenecks and limiting resources to movement through the seaport?
- Why are operations not completed by the required time?
- What are the implications if certain seaport resources are constrained or made available?
- What is the port throughput capability, given explicit assumptions on assets, resources, and scenarios?

PORTSIM addresses two modes of operation. The first mode involves embarkation processes and encompasses the major activities of (1) reception, (2) staging, and (3) ship loading. Reception activities include all processes needed to accept cargo items at the entry points to the port and then transport those items to the staging areas. The entry points to the port are the gates for highway entry and interchange yards for railway entry. Staging activities encompass all processes necessary to park cargo items in appropriate locations before ship arrival as well as to inspect individual items. Finally, ship loading activities include all processes needed to call forward the cargo items to the berths when ships arrive and to load them onto the ships by means of the appropriate loading method (i.e., roll-on, roll-off [RORO]; lift-on, lift-off [LOLO]).

The second mode of operation in PORTSIM addresses debarkation processes and encompasses the major activities of (1) ship unloading, (2) staging, and (3) clearance. Ship unloading activities include all processes needed to offload the vessels that arrive at the berths and transport the cargo to the staging areas. Debarkation staging activities are similar to embarkation staging activities and encompass the parking and inspection of cargo items at the port. Finally, clearance activities allow the cargo items to be moved
from the port onto the highway or railway infrastructure. This activity involves the loading of commercial highway assets such as flatbed trucks and chassis and the loading of railway assets such as trains made up of flatcars and boxcars.

3. Previous Model Design and Limitations
PORTSIM was originally developed for the Unix platform running on a Sun workstation. The model includes the following components: (1) a pre-processing user interface that allows for scenario creation and manipulation; (2) a simulation engine that executes all simulation processing activity; and (3) a post-processing user interface that allows users to view statistical results tables, graphs, and charts. In the original model, the pre-processing and simulation processing components were written with the MODSIM programming language [5] and associated Simgraphics libraries for graphical user interface display. The post-processing component utilized the XRT graphics package [6], which is a C language library that allows for professional output of charts and graphs, in conjunction with a PostScript writer, also written in C language, for output of reports and tables. The back end database was stored in ORACLE, and the model utilized embedded SQL within the simulation engine component to retrieve required information from ship and port data tables. Figure 1 illustrates the original design of the PORTSIM model.

This model design was attractive during initial development and had several advantages. First, MODSIM provided a solid foundation for the development of the simulation processing component of the model. MODSIM contained the necessary simulation constructs such as time clock management, event queue handling, and objects with inheritance hierarchy, to prevent duplication of previous efforts and to reduce the overall development cycle. It also allowed for easier reuse of simulation code throughout the model. Finally, it provided an adequate user interface capability for pre-processing that could be quickly generated for scenario manipulation. Another advantage was in the use of embedded SQL for data retrieval from ORACLE. The embedded SQL allowed for very quick and efficient database access calls, reducing the time users would need to wait when running the model. A final advantage of the initial design was in the use of a PostScript writer for output reports and tables. This code could produce hundreds of pages of formatted output in a very short time.

These advantages in model design led to the initial success of PORTSIM. However, as the user base grew and the model was fielded to a larger number of locations, several limitations of the model design became apparent. Some of the original design advantages that were achieved when executing only on the Sun workstation became disadvantages when the fielding requirements for the model changed to include other machines, such as Windows NT machines, SGI workstations, and HP workstations. The limitations were based on the fact that portability of the model code became the primary focus of the users. The original model design had limited portability, which resulted in extended development timelines to port the code from machine to machine. The limitations consisted of the following items:

- The embedded SQL code became difficult to port from machine to machine, while at the same time requiring a tight linkage between the simulation model and the
back end ORACLE database. Users on different platforms requested the ability to utilize different database management systems, but the embedded SQL code required significant effort to port and made responding to user requests in a timely manner difficult.

- The PostScript writer C code for output reports and tables, while very efficient, was difficult to port from one machine to another.

- The XRT graphics C code, utilized for output charts and graphs, was also difficult to port from one machine to another.

After receiving user feedback, and experiencing some frustration in attempting to respond to the user feedback, it was determined that the model design needed to be modified to better accommodate new user requirements and to make fielding and maintenance of the model an easier process.

4. New and Improved Model Design

The new model design for PORTSIM alleviates the limitations of the previous model design and provides model users with three critical capabilities: (1) utilize the user-specified database management system of choice; (2) present post-processing graphics output with a consistent look and feel on varying platforms; and (3) provide greater flexibility in transmitting model results in the form of graphs, tables, reports, and charts, to other individuals who are not model users, but who require the results of model analysis. The new model design for PORTSIM is presented in Figure 2.

4.1 Database Access

To meet the requirement of easily utilizing user-specified database management systems, the embedded SQL approach for accessing ORACLE needed to be eliminated. Users desired the capability to be able to work with Microsoft Access and Microsoft SQL Server in addition to the existing capability to work with ORACLE. The design approach that was selected to solve this problem was to utilize the Java programming language [7,8] with the Java Database Connectivity (JDBC) library. Java provides the distinct advantage that the code could be written once and could be ported to various machines and platforms without needing to recompile or modify the source code. In addition, JDBC allows users to connect to a wide variety of database management systems, including those listed above, without modifying the source code. The approach involved developing a client-server model that consists of a simulation client process written in MODSIM and a separate database server process written in Java. The database server process does not require a user interface, and its purpose is to wait for database requests and respond by fulfilling the request with database information from the user-specified database system. Communication between the simulation client process and the database server is completed through socket implementation. De-coupling the simulation process from the database server process resulted in a more modular design and allowed for greater flexibility for users because it addressed the new requirement effectively. The database server can be located on a local machine or as a remote server that can be accessed over a network. Performance of database retrieval decreased marginally, due to the additional socket communication as well as the JDBC overhead, but the model
performance was not significantly impacted, and the benefit of new capability to users and ease with which the model could be fielded far outweighed the decrease in performance.

4.2 Graphics Output
A similar approach was taken to decrease portability issues with the reporting and graphing capabilities. The XRT graphics library usage was replaced with a new Java graphic output server that responds to simulation process requests for reports or graphs. Communication between the MODSIM simulation client and the Java graphics server is also completed through socket communication. Rather than build a new graphing and reporting library, a third-party software package named JClass Chart from KL Group [9] was selected. This graphics package, written in Java, provides PORTSIM with professionally formatted output that can easily be ported and supports the added benefit of interactive input from model users. The interactive feature of this graphics library is an added capability that did not exist in the previous design. This feature allows users to query the graphs and reports directly to obtain additional information. Performance of graphics reporting capabilities utilizing the new model design also decreased marginally, but feedback from PORTSIM users has verified that the increased capability and flexibility exceed the marginal reduction in speed of execution.

4.3 Pre-processing User Interface and Simulation Engine
It should be noted that the pre-processing user interface component and the simulation engine component of the PORTSIM simulation model remain in MODSIM under the new design. MODSIM is a simulation language that ports fairly easily from machine to machine, but recompilation on the target machine is required. The new design does not eliminate all issues in facilitating fielding on multiple platforms, but greatly reduces the level of effort required to complete the port. The new design does utilize MODSIM more appropriately, primarily for the simulation engine constructs and capabilities, which are the overall strengths of MODSIM.

5. Conclusions
Fielding and maintenance of large simulation models can become very labor intensive. Demands are high to produce new model capabilities and to deliver these capabilities to users that utilize many different computing machines and platforms. These demands are often accompanied by very tight deadlines. Flexible model design that reduces the amount of development time required to meet these fielding and maintenance requirements is critical to meeting the requirements of users in the most efficient manner. PORTSIM is an example of how the appropriate model design can lead to cost and labor savings when fielding to large numbers of sites.

6. Acknowledgment
This work was supported under military interdepartmental purchase request from the U.S. Department of Defense, Military Traffic Management Command Transportation Engineering Agency, through the U.S. Department of Energy contract W-31-109-Eng-38.
7. References


ORIGINAL PORTSIM DESIGN

PORTSIM

Pre-processing Component (GUI)
Simulation Component

MODSIM

Embedded SQL (C code)
XRT Graphics (C code)
PostScript Writer (C code)

ORACLE Database

- Port Data
- Ship Data

Graphs

- Utilization Graphs
- Throughput Graphs
- Closure Graphs

Reports

- Cargo Reports
- Port Reports
- Ship Reports
- Scenario Reports

Figure 1
NEW PORTSIM DESIGN

PORTSIM

Pre-processing Component (GUI)
Simulation Component

MODSIM

Socket Communication

Database Server
Java
JDBC Java Code

ORACLE Database
- Port Data
- Ship Data

Socket Communication

Graph Server
Java

Graphs and Reports
- All Graphs
- All Reports
- All Tables

Figure 2